Path Validation Testing for PKI Client Protection Profiles

NIST intends to use the Public Key Interoperability Test Suite (PKITS) as the basis for a series of Protection Profiles. Each of the Protection Profiles will consist of a core set of functionality that all applications must implement and a set of packages that specify optional functionality.

This document describes the initial draft core functional requirements as well as the functional requirements for each of the proposed packages. Each of the tests from PKITS is listed below along with an indication of which applications should run each test. While most of the tests are to be run on all applications, many of the tests only need to be run on applications that implement a specified optional feature (e.g., processing name constraints for for **uniformResourceIdentifier** name form).

In order to claim conformance to one of the Protection Profiles, all applications must implement at least the following features:

1. Must be able to process certificates with RSA subject public keys and RSA PKCS #1 with SHA-1 signatures.
2. Must be able to compare **issuer** and **subject** names with attribute values encoded using **PrintableString** using the name comparison rules specified in section 4.1.2.4 of RFC 3280.
3. Must be able to process **notBefore** and **notAfter** fields in certificates and **thisUpdate** and **nextUpdate** fields in CRLs in both **UTCTime** and **GeneralizedTime**.
4. Must be able to process the **keyUsage** and **basicConstraints** extensions.
5. Must reject certificates and CRLs that include unrecognized critical extensions or critical extensions with unrecognized fields.
6. Must be able to process the **certificatePolicies** extension (but do not need to be able to process **anyPolicy** as a special policy OID). Applications must be able to process **certificatePolicies** extensions in which **policyQualifiers** are present, but the qualifiers may be ignored.
7. Must be able to process the **requireExplicitPolicy** field of the **policyConstraints** extension.
8. Must be able to set the initial-policy-set and initial-explicit-policy (NOTE: It is OK if the application is designed such that *initial-explicit-policy* is set if and only if *initial-policy-set* is not *any-policy*.) In other words, it must be possible for the user to specify, by some means, a set of policies and have path validation succeed only if the path is valid under at least one of those policies.
9. Must be able to configure the application so that the path will only be deemed valid if none of the certificates in the path have been revoked (i.e., if the application can not obtain valid up-to-date certificate status information for each certificate in the path, it must reject the path or at least provide the user with a warning that the status can not

be determined).

10. Must be able to process complete CRLs and segmented CRLs (where the certificate includes a **cRLDistributionPoints** extension with a **distributionPoint** specified as a **fullName** and the CRL includes an **issuingDistributionPoint** extension with a **distributionPoint** specified as a **fullName**).
11. Must be able to process CRLs which contain an **issuingDistributionPoint** extension in which either **onlyContainsCACerts** or **onlyContainsUserCerts** is set to true.

Applications that implement that above functionality will be considered acceptable for use in enterprise PKIs. That is, applications that wish to make use of the organization's PKI for strictly internal applications should have sufficient functionality. The above functionality is *not* considered sufficient for applications that need to leverage external PKIs (e.g., validate paths "through" a Bridge CA).

In addition to the core functionality, the Protection Profiles will include the following optional packages. In order to claim conformance to a package, the application must implement all of functionality specified in that package. However, applications that implement a subset of the functionality in a package will be able to specify within their Security Targets what functionality they implement, and that subset will be tested as indicated in the table below.

I. Name Constraints

Must be able to process the **nameConstraints** extension and be able to process name constraints for the **directoryName** and **rfc822Name** name forms.

II. Policy Mapping

Must be able to process the **policyMappings** extension and the **inhibitPolicyMapping** field of the **policyConstraints** extension, and must allow *initial-policy-mapping-inhibit* to be set.

III. anyPolicy

Must be able to process the special policy **anyPolicy** as specified in RFC 3280 and the **inhibitAnyPolicy** extension, and must allow *initial-inhibit-any-policy* to be set.

IV. Delta-CRLs

Must be able to process delta-CRLs.

V. Distribution Points

Must be able to process the **onlySomeReasons** fields of the **issuingDistributionPoint** extension.

VI.Indirect CRLs

Must be able to process indirect CRLs, including the **cRLIssuer** field of the **cRLDistributionPoints** extension, the **indirectCRL** field of the **issuingDistributionPoint** extension, and the **certificateIssuer** CRL entry extension.

VII.DSA

Must be able to process DSA public keys and signatures, including public keys in which the parameters have been inherited.

Applications may also indicate support for the following optional features which are not included in any package:

1. The ability to process name constraints for the **uniformResourceIdentifier** and **dNSName** name form.
2. The ability to process **cRLDistributionPoints** extensions and **issuingDistributionPoint** extensions that include a **distributionPoint** specified as **nameRelativeToIssuer**.

Applications that implement the Name Constraints, Policy Mapping, and anyPolicy packages in addition to the core functionality may indicate that they are Bridge-enabled. Applications that implement the Distribution Points and Indirect CRLs packages may indicate that they support Advanced CRL processing.

Below is a list of each test in PKITS along with an indication which application should run the test, depending on the set of functionality implemented by that application.

Test	***Who should run test***
4.1.1 Valid Signatures Test1	All.
4.1.2 Invalid CA Signature Test2	All.
4.1.3 Invalid EE Signature Test3	All.
4.1.4 Valid DSA Signatures Test4	All. Applications that can not verify DSA signatures must reject the path.
4.1.5 Valid DSA Parameter Inheritance Test5	Run only if application can verify DSA signatures and parameter inheritance.
4.1.6 Invalid DSA Signature Test6	Run only if application can verify DSA signatures
4.2.1 Invalid CA notBefore Date Test1	All.
4.2.2 Invalid EE notBefore Date Test2	All.

Test	*Who should run test*
4.2.3 Valid pre2000 UTC notBefore Date Test3	All.
4.2.4 Valid GeneralizedTime notBefore Date Test4	All.
4.2.5 Invalid CA notAfter Date Test5	All.
4.2.6 Invalid EE notAfter Date Test6	All.
4.2.7 Invalid pre2000 UTC EE notAfter Date Test7	All.
4.2.8 Valid GeneralizedTime notAfter Date Test8	All.
4.3.1 Invalid Name Chaining EE Test1	All.
4.3.2 Invalid Name Chaining Order Test2	All.
4.3.3 Invalid Name Chaining Whitespace Test3	All.
4.3.4 Valid Name Chaining Whitespace Test4	All.
4.3.5 Valid Name Chaining Capitalization Test5	All.
4.3.6 Valid Name Chaining UIDs Test6	All.
4.3.7 Valid RFC3280 Mandatory Attribute Types Test7	This test does not need to be run.
4.3.8 Valid RFC3280 Optional Attribute Types Test8	This test does not need to be run.
4.3.9 Valid UTF8String Encoded Names Test9	All.
4.3.10 Valid Rollover from PrintableString to UTF8String Test10	This test does not need to be run.
4.3.11 Valid UTF8String Case Insensitive Match Test11	This test does not need to be run.
4.4.1 Missing CRL Test1	All.

Test	*Who should run test*
4.4.2 Invalid Revoked CA Test2	All.
4.4.3 Invalid Revoked EE Test3	All.
4.4.4. Invalid Bad CRL Signature Test4	All.
4.4.5 Invalid Bad CRL Issuer Name Test5	All.
4.4.6 Invalid Wrong CRL Test6	All.
4.4.7 Valid Two CRLs Test7	All.
4.4.8 Invalid Unknown CRL Entry Extension Test8	All.
4.4.9 Invalid Unknown CRL Extension Test9	All.
4.4.10 Invalid Unknown CRL Extension Test10	All.
4.4.11 Invalid Old CRL nextUpdate Test11	All.
4.4.12 Invalid pre2000 CRL nextUpdate Tesst12	All.
4.4.13 Valid GeneralizedTime CRL nextUpdate Test13	All.
4.4.14 Valid Negative Serial Number Test14	All.
4.4.15 Invalid Negative Serial Number Test15	All.
4.4.16 Valid Long Serial Number Test16	All.
4.4.17 Valid Long Serial Number Test17	All.
4.4.18 Valid Long Serial Number Test18	All.
4.4.19 Valid Separate Certificate and CRL Keys Test19	All.
4.4.20 Invalid Separate Certificate and CRL Keys Test20	All.
4.4.21 Invalid Separate Certificate and CRL Keys Test21	All.

Test	*Who should run test*
4.5.1 Valid Basic Self-Issued Old With New Test1	All.
4.5.2 Invalid Basic Self-Issued Old With New Test2	All.
4.5.3 Valid Basic Self-Issued New With Old Test3	All.
4.5.4 Valid Basic Self-Issued New With Old Test4	All.
4.5.5 Invalid Basic Self-Issued New With Old Test5	All.
4.5.6 Valid Basic Self-Issued CRL Signing Key Test6	All.
4.5.7 Invalid Basic Self-Issued CRL Signing Key Test7	All.
4.5.8 Invalid Basic Self-Issued CRL Signing Key Test8	All.
4.6.1 Invalid Missing basicConstraints Test1	All.
4.6.2 Invalid cA False Test2	All.
4.6.3 Invalid cA False Test3	All.
4.6.4 Valid basicConstraints Not Critical Test4	All.
4.6.5 Invalid pathLenConstraint Test5	All.
4.6.6 Invalid pathLenConstraint Test6	All.
4.6.7 Valid pathLenConstraint Test7	All.
4.6.8 Valid pathLenConstraint Test8	All.
4.6.9 Invalid pathLenConstraint Test9	All.
4.6.10 Invalid pathLenConstraint Test10	All.
4.6.11 Invalid pathLenConstraint Test11	All.

Test	*Who should run test*
4.6.12 Invalid pathLenConstraint Test12	All.
4.6.13 Valid pathLenConstraint Test13	All.
4.6.14 Valid pathLenConstraint Test14	All.
4.6.15 Valid Self-Issued pathLenConstraint Test15	All.
4.6.16 Invalid Self-Issued pathLenConstraint Test16	All.
4.6.17 Valid Self-Issued pathLenConstraint Test17	All.
4.7.1 Invalid keyUsage Critical keyCertSign False Test1	All.
4.7.2 Invalid keyUsage Not Critical keyCertSign False Test2	All.
4.7.3 Valid keyUsage Not Critical Test3	All.
4.7.4 Invalid keyUsage Critical cRLSign False Test4	All.
4.7.5 Invalid keyUsage Not Critical cRLSign False Test5	All.
4.8.1 All Certificates Same Policy Test1, subtest 1	Run if application can be configured as specified (i.e., if *initial-policy-set* can be *any-policy* when *initial-explicit-policy* is set).
4.8.1 All Certificates Same Policy Test1, subtest 2	All.
4.8.1 All Certificates Same Policy Test1, subtest 3	All.
4.8.1 All Certificates Same Policy Test1, subtest 4	All.
4.8.2 All Certificates No Policies Test2, subtest 1	All.
4.8.2 All Certificates No Policies Test2, subtest 2	All. (*initial-policy-set* may be set to {NIST-test-policy-1, NIST-test-policy-2, NIST-test-policy-3, NIST-test-policy-4, NIST-test-policy-5, NIST-test-policy-6} if it can not be set to *any-policy*).

Test	*Who should run test*
4.8.3 Different Policies Test3, subtest 1	All.
4.8.3 Different Policies Test3, subtest 2	Run if application can be configured as specified (i.e., if *initial-policy-set* can be *any-policy* when *initial-explicit-policy* is set).
4.8.3 Different Policies Test3, subtest 3	All.
4.8.4 Different Policies Test4	All.
4.8.5 Different Policies Test5	All.
4.8.6 Overlapping Policies Test6, subtest 1	All.
4.8.6 Overlapping Policies Test6, subtest 2	All.
4.8.6 Overlapping Policies Test6, subtest 3	All.
4.8.7 Different Policies Test7	All.
4.8.8 Different Policies Test8	All.
4.8.9 Different Policies Test9	All.
4.8.10 All Certificates Same Policies Test10, subtest 1	All.
4.8.10 All Certificates Same Policies Test10, subtest 2	All.
4.8.10 All Certificates Same Policies Test10, subtest 3	All.
4.8.11 All Certificates AnyPolicy Test11, subtest 1	This subtest does not need to be run.
4.8.11 All Certificates AnyPolicy Test11, subtest 2	Run if application can process the special policy **anyPolicy**.
4.8.12 Different Policies Test12	All.
4.8.13 All Certificates Same Policies Test13, subtest 1	All.
4.8.13 All Certificates Same Policies Test13, subtest 2	All.
4.8.13 All Certificates Same Policies Test13, subtest 3	All.

Test	*Who should run test*
4.8.14 AnyPolicy Test14, subtest 1	Run if application can process the special policy **anyPolicy**.
4.8.14 AnyPolicy Test14, subtest 2	Run if application can process the special policy **anyPolicy**.
4.8.15 User Notice Qualifier Test15	This test does not need to be run.
4.8.16 User Notice Qualifier Test16	This test does not need to be run.
4.8.17 User Notice Qualifier Test17	This test does not need to be run.
4.8.18 User Notice Qualifier Test18, subtest 1	This subtest does not need to be run.
4.8.18 User Notice Qualifier Test18, subtest 2	This subtest does not need to be run.
4.8.19 User Notice Qualifier Test19	This test does not need to be run.
4.8.20 CPS Pointer Qualifier Test20	All. Test should be run with *initial-explicit-policy* set (*initial-policy-set* may be set to {NIST-test-policy-1, NIST-test-policy-2, NIST-test-policy-3, NIST-test-policy-4, NIST-test-policy-5, NIST-test-policy-6} if it can not be set to *any-policy*).
4.9.1 Valid RequireExplicitPolicy Test1	All.
4.9.2 Valid RequireExplicitPolicy Test2	All.
4.9.3 Invalid RequireExplicitPolicy Test3	All.
4.9.4 Valid RequireExplicitPolicy Test4	All.
4.9.5 Invalid RequireExplicitPolicy Test5	All.
4.9.6 Valid Self-Issued requireExplicitPolicy Test6	All.
4.9.7 Invalid Self-Issued requireExplicitPolicy Test7	All.
4.9.8 Invalid Self-Issued requireExplicitPolicy Test8	All.

Test	*Who should run test*
4.10.1 Valid Policy Mapping Test1, subtest 1	All. Applications that can not process the **policyMappings** extension should reject the path (When testing an application that does not process the **policyMappings** extension, the default settings should be used (i.e., *initial-policy-set = any-policy*)).
4.10.1 Valid Policy Mapping Test1, subtest 2	Run if application can process the **policyMappings** extension.
4.10.1 Valid Policy Mapping Test1, subtest 3	Run if *initial-policy-mapping-inhibit* can be set.
4.10.2 Invalid Policy Mapping Test2, subtest 1	Run if application can process the **policyMappings** extension.
4.10.2 Invalid Policy Mapping Test2, subtest 2	Run if *initial-policy-mapping-inhibit* can be set.
4.10.3 Valid Policy Mapping Test3, subtest 1	Run if application can process the **policyMappings** extension.
4.10.3 Valid Policy Mapping Test3, subtest 2	Run if application can process the **policyMappings** extension.
4.10.4 Invalid Policy Mapping Test4	Run if application can process the **policyMappings** extension.
4.10.5 Valid Policy Mapping Test5, subtest 1	Run if application can process the **policyMappings** extension.
4.10.5 Valid Policy Mapping Test5, subtest 2	Run if application can process the **policyMappings** extension.
4.10.6 Valid Policy Mapping Test6, subtest 1	Run if application can process the **policyMappings** extension.
4.10.6 Valid Policy Mapping Test6, subtest 2	Run if application can process the **policyMappings** extension.
4.10.7 Invalid Mapping From anyPolicy Test7	Run if application can process the **policyMappings** extension and the special policy **anyPolicy**.
4.10.8 Invalid Mapping To anyPolicy Test8	Run if application can process the **policyMappings** extension and the special policy **anyPolicy**.
4.10.9 Valid Policy Mapping Test9	This test does not need to be run.
4.10.10 Invalid Policy Mapping Test10	All.
4.10.11 Valid Policy Mapping Test11	All.

Test	*Who should run test*
4.10.12 Valid Policy Mapping Test12, subtest 1	Run if application can process the **policyMappings** extension. It is irrelevant whether the user notice is displayed.
4.10.12 Valid Policy Mapping Test12, subtest 2	Run if application can process the **policyMappings** extension and the special policy **anyPolicy**. It is irrelevant whether the user notice is displayed.
4.10.13 Valid Policy Mapping Test13	This test does not need to be run.
4.10.14 Valid Policy Mapping Test14	This test does not need to be run.
4.11.1 Invalid inhibitPolicyMapping Test1	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.2 Valid inhibitPolicyMapping Test2	All. Applications that can not process the **inhibitPolicyMapping** field in the **policyConstraints** extension should reject the path.
4.11.3 Invalid inhibitPolicyMapping Test3	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.4 Valid inhibitPolicyMapping Test4	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.5 Invalid inhibitPolicyMapping Test5	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.6 Invalid inhibitPolicyMapping Test6	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.7 Valid Self-Issued inhibitPolicyMapping Test7	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.8 Invalid Self-Issued inhibitPolicyMapping Test8	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.9 Invalid Self-Issued inhibitPolicyMapping Test9	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.

Test	*Who should run test*
4.11.10 Invalid Self-Issued inhibitPolicyMapping Test10	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.11.11 Invalid Self-Issued inhibitPolicyMapping Test11	Run if application can process the **inhibitPolicyMapping** field in the **policyConstraints** extension.
4.12.1 Invalid inhibitAnyPolicy Test1	Run if application can process the **inhibitAnyPolicy** extension.
4.12.2 Valid inhibitAnyPolicy Test2	All. Applications that can not process the **inhibitAnyPolicy** extension should reject the path.
4.12.3 inhibitAnyPolicy Test3, subtest 1	Run if application can process the **inhibitAnyPolicy** extension.
4.12.3 inhibitAnyPolicy Test3, subtest 2	Run if *initial-inhibit-any-policy* can be set.
4.12.4 Invalid inhibitAnyPolicy Test4	Run if application can process the **inhibitAnyPolicy** extension.
4.12.5 Invalid inhibitAnyPolicy Test5	Run if application can process the **inhibitAnyPolicy** extension.
4.12.6 Invalid inhibitAnyPolicy Test6	Run if application can process the **inhibitAnyPolicy** extension.
4.12.7 Valid Self-Issued inhibitAnyPolicy Test7	Run if application can process the **inhibitAnyPolicy** extension.
4.12.8 Invalid Self-Issued inhibitAnyPolicy Test8	Run if application can process the **inhibitAnyPolicy** extension.
4.12.9 Valid Self-Issued inhibitAnyPolicy Test9	Run if application can process the **inhibitAnyPolicy** extension.
4.12.10 Invalid Self-Issued inhibitAnyPolicy Test10	Run if application can process the **inhibitAnyPolicy** extension.
4.13.1 Valid DN nameConstraints Test1	All. Applications that can not process name constraints for the **directoryName** form should reject the path.
4.13.2 Invalid DN nameConstraints Test2	Run if application can process name constraints for the **directoryName** form.
4.13.3 Invalid DN nameConstraints Test3	Run if application can process name constraints for the **directoryName** form.
4.13.4 Valid DN nameConstraints Test4	Run if application can process name constraints for the **directoryName** form.

Test	*Who should run test*
4.13.5 Valid DN nameConstraints Test5	Run if application can process name constraints for the **directoryName** form.
4.13.6 Valid DN nameConstraints Test6	Run if application can process name constraints for the **directoryName** form.
4.13.7 Invalid DN nameConstraints Test7	Run if application can process name constraints for the **directoryName** form.
4.13.8 Invalid DN nameConstraints Test8	Run if application can process name constraints for the **directoryName** form.
4.13.9 Invalid DN nameConstraints Test9	Run if application can process name constraints for the **directoryName** form.
4.13.10 Invalid DN nameConstraints Test10	Run if application can process name constraints for the **directoryName** form.
4.13.11 Valid DN nameConstraints Test11	Run if application can process name constraints for the **directoryName** form.
4.13.12 Invalid DN nameConstraints Test12	Run if application can process name constraints for the **directoryName** form.
4.13.13 Invalid DN nameConstraints Test13	Run if application can process name constraints for the **directoryName** form.
4.13.14 Valid DN nameConstraints Test14	Run if application can process name constraints for the **directoryName** form.
4.13.15 Invalid DN nameConstraints Test15	Run if application can process name constraints for the **directoryName** form.
4.13.16 Invalid DN nameConstraints Test16	Run if application can process name constraints for the **directoryName** form.
4.13.17 Invalid DN nameConstraints Test17	Run if application can process name constraints for the **directoryName** form.
4.13.18 Valid DN nameConstraints Test18	Run if application can process name constraints for the **directoryName** form.
4.13.19 Valid Self-Issued DN nameConstraints Test19	Run if application can process name constraints for the **directoryName** form.
4.13.20 Invalid Self-Issued DN nameConstraints Test20	Run if application can process name constraints for the **directoryName** form.
4.13.21 Valid RFC822 nameConstraints Test21	Run if application can process the **nameConstraints** extension (for any name form). Applications that can not process name constraints for the **rfc822Name** form should reject the path.

Test	*Who should run test*
4.13.22 Invalid RFC822 nameConstraints Test22	Run if application can process name constraints for the **rfc822Name** form.
4.13.23 Valid RFC822 nameConstraints Test23	Run if application can process name constraints for the **rfc822Name** form.
4.13.24 Invalid RFC822 nameConstraints Test24	Run if application can process name constraints for the **rfc822Name** form.
4.13.25 Valid RFC822 nameConstraints Test25	Run if application can process name constraints for the **rfc822Name** form.
4.13.26 Invalid RFC822 nameConstraints Test26	Run if application can process name constraints for the **rfc822Name** form.
4.13.27 Valid DN and RFC822 nameConstraints Test27	Run if application can process name constraints for both the **directoryName** form and the **rfc822Name** form.
4.13.28 Invalid DN and RFC822 nameConstraints Test28	Run if application can process name constraints for both the **directoryName** form and the **rfc822Name** form.
4.13.29 Invalid DN and RFC822 nameConstraints Test29	Run if application can process name constraints for both the **directoryName** form and the **rfc822Name** form.
4.13.30 Valid DNS nameConstraints Test30	Run if application can process the **nameConstraints** extension (for any name form). Applications that can not process name constraints for the **dNSName** form should reject the path.
4.13.31 Invalid DNS nameConstraints Test31	Run if application can process name constraints for the **dNSName** form.
4.13.32 Valid DNS nameConstraints Test32	Run if application can process name constraints for the **dNSName** form.
4.13.33 Invalid DNS nameConstraints Test33	Run if application can process name constraints for the **dNSName** form.
4.13.34 Valid URI nameConstraints Test34	Run if application can process the **nameConstraints** extension (for any name form). Applications that can not process name constraints for the **uniformResourceIdentifier** name form should reject the path.
4.13.35 Invalid URI nameConstraints Test35	Run if application can process name constraints for the **uniformResourceIdentifier** name form.
4.13.36 Valid URI nameConstraints Test36	Run if application can process name constraints for the **uniformResourceIdentifier** name form.

Test	*Who should run test*
4.13.37 Invalid URI nameConstraints Test37	Run if application can process name constraints for the **uniformResourceIdentifier** name form.
4.13.38 Invalid DNS nameConstraints Test38	Run if application can process name constraints for the **dNSName** form.
4.14.1 Valid distributionPoint Test1	All.
4.14.2 Invalid distributionPoint Test2	All.
4.14.3 Invalid distributionPoint Test3	All.
4.14.4 Valid distributionPoint Test4	All. Applications that can not process **cRLDistributionPoints** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer** should reject the path (or issue a warning that certificate status can not be determined).
4.14.5 Valid distributionPoint Test5	All. Applications that can not process **cRLDistributionPoints** extensions or **issuingDistributionPoint** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer** should reject the path (or issue a warning that certificate status can not be determined).
4.14.6 Invalid distributionPoint Test6	Run if application can process **cRLDistributionPoints** extensions and **issuingDistributionPoint** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer**.
4.14.7 Valid distributionPoint Test7	Run if application can process **issuingDistributionPoint** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer**.
4.14.8 Invalid distributionPoint Test8	Run if application can process **issuingDistributionPoint** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer**.
4.14.9 Invalid distributionPoint Test9	All.

Test	*Who should run test*
4.14.10 Valid No issuingDistributionPoint Test10	All.
4.14.11 Invalid onlyContainsUserCerts CRL Test11	All.
4.14.12 Invalid onlyContainsCACerts CRL Test12	All.
4.14.13 Invalid onlyContainsCACerts CRL Test13	All.
4.14.14 Invalid onlyContainsAttributeCerts Test14	All.
4.14.15 Invalid onlySomeReasons Test15	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.16 Invalid onlySomeReasons Test16	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.17 Invalid onlySomeReasons Test17	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.18 Valid onlySomeReasons Test18	All. Applications that can not process the **onlySomeReasons** field of the **issuingDistributionPoint** extension should reject the path (or issue a warning that certificate status can not be determined).
4.14.19 Valid onlySomeReasons Test19	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.20 Invalid onlySomeReasons Test20	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.21 Invalid onlySomeReasons Test21	Run if application can process the **onlySomeReasons** field of the **issuingDistributionPoint** extension.
4.14.22 Valid IDP with indirectCRL Test22	Run if application can process indirect CRLs.

Test	*Who should run test*
4.14.23 Invalid IDP with indirectCRL Test23	Run if application can process indirect CRLs.
4.14.24 Valid IDP with indirectCRL Test24	All. Applications that can not process indirect CRLs should reject the path (or issue a warning that certificate status can not be determined).
4.14.25 Valid IDP with indirectCRL Test25	Run if application can process indirect CRLs.
4.14.26 Invalid IDP with indirectCRL Test26	Run if application can process indirect CRLs.
4.14.27 Invalid cRLIssuer Test27	Run if application can process indirect CRLs.
4.14.28 Valid cRLIssuer Test28	Run if application can process indirect CRLs.
4.14.29 Valid cRLIssuer Test29	Run if application can process indirect CRLs and **cRLDistributionPoints** extensions that include a **distributionPoint** that is specified as **nameRelativeToCRLIssuer**.
4.14.30 Valid cRLIssuer Test30	This test does not need to be run.
4.14.31 Invalid cRLIssuer Test31	Run if application can process indirect CRLs.
4.14.32 Invalid cRLIssuer Test32	Run if application can process indirect CRLs.
4.14.33 Valid cRLIssuer Test33	Run if application can process indirect CRLs.
4.14.34 Invalid cRLIssuer Test34	Run if application can process indirect CRLs.
4.14.35 Invalid cRLIssuer Test35	Run if application can process indirect CRLs.
4.15.1 Invalid deltaCRLIndicator No Base Test1	All.
4.15.2 Valid delta-CRL Test2	Run if application can process delta-CRLs.
4.15.3 Invalid delta-CRL Test3	Run if application can process delta-CRLs.
4.15.4 Invalid delta-CRL Test4	Run if application can process delta-CRLs.
4.15.5 Valid delta-CRL Test5	Run if application can process delta-CRLs.
4.15.6 Invalid delta-CRL Test6	Run if application can process delta-CRLs.
4.15.7 Valid delta-CRL Test7	Run if application can process delta-CRLs.
4.15.8 Valid delta-CRL Test8	Run if application can process delta-CRLs.
4.15.9 Invalid delta-CRL Test9	Run if application can process delta-CRLs.
4.15.10 Invalid delta-CRL Test10	Run if application can process delta-CRLs.

Test	*Who should run test*
4.16.1 Valid Unknown Not Critical Certificate Extension Test1	All.
4.16.2 Invalid Unknown Critical Certificate Extension Test2	All.